```
                  *H RML 380Z MONITOR V 2.3 R

                  ;-----------------------------------------------------------
                  ;************ R.M.L. CASSETTE OPERATING SYSTEM ***********
                  ;************  MONITOR VERSION 2.3 R  REV. 0   ***********
                  ;-----------------------------------------------------------
                  ;-----------------------------------------------------------

                  ;         'BY VIEWING NATURE, NATURE'S HANDMAID, ART,
                  ;          MAKES MIGHTY THINGS FROM SMALL BEGINNINGS GROW.'
                  ;                                            DRYDEN

0000                         ORG     0
       1000       NXTROM     EQU     1000H
       4000       RAM        EQU     4000H

0000   C33703     MONST:     JP      BEGIN          ;RST 00H (& POWER ON RESET)
0003   C37103                JP      CONTC          ;ADDR FOR REENTER CMD
0006   544F       MSGM:      DEFM    'TO'
0008   C32E40                JP      RST8           ;RST 08H
000B   42524541   MSGB:      DEFM    'BREAK'
0010   C33140                JP      RST10          ;RST 10H
0013   3F455252   MSGE:      DEFM    '?ERR?'
0018   C33440                JP      RST18          ;RST 18H
001B   46495253   MSGF:      DEFM    'FIRST'
0020   C33740                JP      RST20          ;RST 20H
0023   53544152   MSGS:      DEFM    'START'
0028   C38500                JP      RCALL          ;RST 28H (RELATIVE CALL)
002B   4C415354   MSGL:      DEFM    'LAST'
002F   FF                    DEFB    -1
0030   1817                  JR      TRAP           ;RST 30H (TRAP CALL)

                  ;32H - THE NEXT TWO BYTES OF CODE ENABLE USER PROGRAMS TO
                  ;FIND OUT WHERE THEY HAVE BEEN LOADED, MAKING IT POSSIBLE TO
                  ;WRITE POSITION INDEPENDENT CODE (PIC).  THEY SHOULD BE
                  ;CALLED ('CALL 32H'); ON RETURN OLD HL IS ON THE STACK
                  ;AND HL CONTAINS THE ADDR OF THE CURRENT INSTRUCTION
                  ;I.E. THE ONE AFTER THE CALL.

0032   E3                    EX      (SP),HL
0033   E9                    JP      (HL)

0034   57495448   MSGP:      DEFM    'WITH'

                  ;BREAKPOINT - COME HERE AFTER RST 38H (OFFH)
                  ;(INTERRUPTS IN MODE 1 ALSO COME HERE)
                  ;OLD PC ON THE STACK IS DECREMENTED, TO POINT
                  ;TO THE LOCATION OF THE BREAKPOINT AND THIS ADDRESS
                  ;IS ALSO PUT IN (MPTR)

0038   E3         BREAK:     EX      (SP),HL        ;HL <- ADDR OF NEXT INSTR
0039   2B                    DEC     HL             ;POINT TO BREAK POINT
003A   220740                LD      (MPTR),HL
003D   E3                    EX      (SP),HL
```

```
003E  E5                     PUSH    HL
003F  F5                     PUSH    AF
0040  210B00                 LD      HL,MSGB          ;ISSUE BREAK MESSAGE
0043  CDCC05                 CALL    MOUT
0046  C3D600                 JP      FPM              ;GO TO FP WITH HL/AF ON STACK

                     ;TRAP - A TRAP INSTRUCTION ('EMT'), SIMULATED BY RST 30H,
                     ;IS A TWO BYTE INSTRUCTION EQUIVALENT TO 'CALL'.  THE SECOND
                     ;BYTE CONTAINS THE TRAP CODE OF THE DESIRED TARGET ROUTINE,
                     ;AS FOLLOWS:
                     ;
                     ;         CODE            TARGET
                     ;
                     ;         -VE             ?ERR?
                     ;          0              MONITOR RESTART
                     ;         1-10            TRANSFER VECTORS ('VTV' - 'IN3')
                     ;         11-24           MONITOR ROUTINES IN 'TRTBL'
                     ;         25-127          JUMP THROUGH 'TRAPX'

0049  E5             TRAP:   PUSH    HL               ;SAVE HL AND AF
004A  F5                     PUSH    AF
004B  CD9200                 CALL    LDRET            ;HL <- RET ADDR, INC RET ADDR
004E  7E                     LD      A,(HL)           ;A <- TRAP CODE
004F  B7                     OR      A
0050  28B1                   JR      Z,MONST+3        ;TRAP 0 - RESTART
0052  FAF305                 JP      M,ERROR          ;-VE TRAP CODE MEANS ERROR

                     ;IF CODE IS > 24(10), JUMP THROUGH TRAP EXTENSION VECTOR 'TRAPX'
                     ;WITH THE CODE IN REG A AND AF, HL AND THE RET ADDR ON THE STACK.

0055  FE19                   CP      25
0057  D23D40                 JP      NC,TRAPX
005A  D60B                   SUB     11
005C  381C                   JR      C,TR2

                     ;CODES 11 TO 24 (DECIMAL, INCLUSIVE) CALL THE MONITOR
                     ;ROUTINES WHOSE ADDRESSES ARE IN THE JUMP TABLE 'TRTBL'

005E  87                     ADD     A,A              ;A <- A * 2
005F  21B507                 LD      HL,TRTBL         ;HL <- ADDR OF DISPATCH TABLE
0062  85                     ADD     A,L              ;HL <- ADDR OF ADDR
0063  6F                     LD      L,A
0064  1802                   JR      TR1              ;AVOID NMI

0066  183A                   JR      NMI              ;LOCATION 66H

0068  7E             TR1:    LD      A,(HL)           ;HL <- DISPATCH ADDR
0069  23                     INC     HL
006A  66                     LD      H,(HL)
006B  6F                     LD      L,A
006C  1814                   JR      TR3

                     ;.CHAN (EMT CHAN) ALLOWS CHANNEL TO BE PASSED IN REGISTER C

```

```
006E   E5              .CHAN:  PUSH   HL
006F   F5                      PUSH   AF
0070   79                      LD     A,C              ;A <- CHANNEL
0071   FE01                    CP     1                ;CHECK THAT IT IS IN THE
0073   3001                    JR     NC,.CHN2         ; RANGE 1 TO 10
0075   FF              .CHN1:  RST    38H              ;ELSE BREAK
0076   D60B            .CHN2:  SUB    11
0078   30FB                    JR     NC,.CHN1

                       ;CODES 1 TO 10 (DECIMAL, INCLUSIVE) REACH THE TRANSFER VECTORS

007A   6F              TR2:    LD     L,A              ;A <- A * 3
007B   87                      ADD    A,A
007C   85                      ADD    A,L
007D   212E40                  LD     HL,IN3+3         ;CALCULATE VECTOR ADDR
0080   85                      ADD    A,L
0081   6F                      LD     L,A
0082   F1              TR3:    POP    AF               ;RESTORE AF AND HL
0083   E3                      EX     (SP),HL
0084   C9                      RET                     ;GO TO ADDR

                       ;RCALL - CALL RELATIVE ('CALR') IS A TWO BYTE PSEUDO-
                       ;INSTRUCTION SIMULATED BY RST 28H.  THE SECOND BYTE
                       ;CONTAINS THE OFFSET TO THE TARGET ADDR WITH THE SAME
                       ;CONVENTION AS THE RELATIVE JUMPS (I.E. TARGET-$-1).
                       ;THIS CODE CONVERTS THE OFFSET TO AN ABSOLUTE ADDR, THEN
                       ;JUMPS THERE, WITH THE UPDATED RETURN ADDR ON THE STACK.

0085   E5              RCALL:  PUSH   HL               ;SAVE HL AND AF
0086   F5                      PUSH   AF
0087   CD9200                  CALL   LDRET            ;HL <- ADDR OF OFFSET
008A   D5                      PUSH   DE               ;SAVE DE
008B   CD6502                  CALL   RTOA             ;HL <- HL + (HL) + 1
008E   D1                      POP    DE               ;RESTORE DE, AF AND HL
008F   F1                      POP    AF
0090   E3                      EX     (SP),HL
0091   C9                      RET                     ;GO TO CALL

                       ;LDRET - LOAD HL WITH PREVIOUS RETURN ADDR, INCREMENT
                       ;RETURN ADDR.  ASSUMES HL AND AF ARE ALREADY ON THE STACK.
                       ;DESTROYS CY FLAG

0092   D5              LDRET:  PUSH   DE
0093   210800                  LD     HL,8             ;HL <- POINTER TO STACK
0096   39                      ADD    HL,SP
0097   5E                      LD     E,(HL)           ;DE <- RETURN ADDR
0098   23                      INC    HL
0099   56                      LD     D,(HL)
009A   13                      INC    DE               ;INC RETURN AND STORE
009B   72                      LD     (HL),D
009C   2B                      DEC    HL
009D   73                      LD     (HL),E
009E   1B                      DEC    DE               ;CORRECT ADDR
009F   EB                      EX     DE,HL            ;HL <- ADDR
```

```
00A0    D1                  POP     DE
00A1    C9                  RET

                ;NMI - COME HERE AFTER NMI CAUSED BY RESET BUTTON OR
                ;SINGLE STEP LOGIC.
                ;IF RESET, WAIT FOR BUTTON RELEASE THEN RESTART MONITOR
                ;ELSE DISABLE SINGLE STEP BIT IN 'MASK' AND CHECK NMI FLAG:
                ;IF (SSFLG) = 'R', JUMP TO USER PROG THROUGH NMIX,
                ;ELSE IF (SSFLG) = 'S' (MONITOR SINGLE STEP) GO TO FP
                ;ELSE TREAT AS RESET BUTTON GLITCH AND RESTART MONITOR

00A2    E5      NMI:        PUSH    HL
00A3    F5                  PUSH    AF
00A4    21FC0F              LD      HL,PORTO            ;DISABLE SINGLE STEP IF SET
00A7    3692                LD      (HL),MKINIT
00A9    23                  INC     HL                  ;POINT TO PORT 1
00AA    23                  INC     HL
00AB    CB56                BIT     RESET,(HL)          ;WAS IT RESET BUTTON?
00AD    2009                JR      NZ,NMI2             ;IF NOT

00AF    CB56    NMI1:       BIT     RESET,(HL)          ;ELSE WAIT FOR BUTTON RELEASE
00B1    28FC                JR      Z,NMI1
00B3    AF                  XOR     A                   ;MAKE SURE SS FLG IS CLEAR
00B4    320440              LD      (SSFLG),A
00B7    C7      NMI1A:      RST     0                   ;GO TO POWER ON ENTRY

00B8    210340  NMI2:       LD      HL,MASK             ;NOT RESET BUTTON,
00BB    CBCE                SET     SINGLE,(HL)         ;(CLEAR SINGLE STEP ANYWAY)
00BD    23                  INC     HL                  ;POINT TO SS FLG
00BE    7E                  LD      A,(HL)              ;TEST FLAG
00BF    3600                LD      (HL),0              ;THEN CLEAR IT
00C1    D652                SUB     'R'
00C3    CA3A40              JP      Z,NMIX              ;IF FLG = 'R' -> NMIX
00C6    3D                  DEC     A                   ;ELSE IF FLG NOT = 'S'
00C7    20EE                JR      NZ,NMI1A            ; TREAT AS GLITCH ON RESET
00C9    F1                  POP     AF                  ;ELSE MUST BE MONITOR SS
00CA    E1                  POP     HL
00CB    1811                JR      FPM1
```

```
                  *H FRONT PANEL ROUTINES

                  ;GETC - GET A CHARACTER FROM KEYBOARD
                  ;WATCH FOR CTRL Z

00CD   F702       GETC:   EMT     KBDIN
00CF   28FC               JR      Z,GETC          ;NO CHAR AVAILABLE
00D1   FE1A               CP      CTRLZ
00D3   C0                 RET     NZ              ;DROP THRO' ON CTRL Z

                  ;HERE BEGINS THE FRONT PANEL EMULATOR
                  ;
                  ;       "AND NOW I SEE WITH EYE SERENE
                  ;        THE VERY PULSE OF THE MACHINE."
                  ;                       WILLIAM WORDSWORTH
                  ;
                  ;SAVE THE CURRENT REGS ON THE MONITOR STACK
                  ;IN THE ORDER:  PC/SP/IY/IX/HL/DE/BC/AF

00D4   E5                 PUSH    HL              ;SAVE AS 'BREAK' DOES
00D5   F5                 PUSH    AF
00D6   D5         FPM:    PUSH    DE              ;ENTRY FROM 'BREAK'
00D7   C5                 PUSH    BC
00D8   F70D               EMT     GRAFIX          ;CLEAR SCREEN
00DA   C1                 POP     BC
00DB   D1                 POP     DE
00DC   F1                 POP     AF
00DD   E1                 POP     HL

00DE   E5         FPM1:   PUSH    HL              ;MAKE ROOM FOR SP
00DF   FDE5               PUSH    IY              ;SAVE ALL
00E1   DDE5               PUSH    IX
00E3   E5         FPM2:   PUSH    HL              ;ENTRY FROM 'SWITCH'
00E4   D5                 PUSH    DE
00E5   C5                 PUSH    BC
00E6   F5                 PUSH    AF
00E7   211000             LD      HL,16           ;CALCULATE OLD SP
00EA   39                 ADD     HL,SP
00EB   54                 LD      D,H             ;DE <- OLD SP
00EC   5D                 LD      E,L
00ED   2B                 DEC     HL
00EE   2B                 DEC     HL
00EF   2B                 DEC     HL
00F0   72                 LD      (HL),D          ;SAVE OLD SP AT (HL)
00F1   2B                 DEC     HL
00F2   73                 LD      (HL),E

                  ;RDIS - DISPLAY THE REGISTERS
                  ;THIS IS THE RE-ENTRY POINT IN FP MODE: ITS ADDR IS PUSHED
                  ;AT 'CMD:' AND MOST FP ROUTINES RETURN HERE WITH A 'RET'.
                  ;IX STARTS BY POINTING TO THE TOP OF THE SAVED REGS (OLD PC)
                  ;AND ENDS 2 BYTES BELOW SAVED AF. (RPTR) IS THE OFFSET
                  ;(RANGE 2-16) FROM THE LATTER POSITION OF IX TO THE CURRENT
                  ;REG. IY IS A POINTER TO THE REG NAME STRING.
```

```
00F3   FD218001          RDIS:   LD      IY,RNSTR        ;IY <- PTR TO REG NAMES
00F7   DD210E00                  LD      IX,14           ;IX <- PTR TO TOP OF SAVED REGS
00FB   DD39                      ADD     IX,SP
00FD   3E21                      LD      A,'!'           ;INDICATE FP MODE
00FF   CD3A05                    CALL    AOUT            ;AFTER CR
0102   F70B                      EMT     OPNWT           ;###
0104   21160A                    LD      HL,0A16H
0107   365E                      LD      (HL),UPARR
0109   218A0B                    LD      HL,0B8AH
010C   365D                      LD      (HL),RARR
010E   21920B                    LD      HL,0B92H
0111   365B                      LD      (HL),LARR
0113   214008                    LD      HL,840H         ;HL <- START OF REG DISPLAY
0116   0608                      LD      B,8
0118   3A0940            RD1:    LD      A,(RPTR)        ;TEST WHICH REG IS POINTED TO
011B   CB2F                      SRA     A               ;(RPTR) = 2*B
011D   B8                        CP      B
011E   3680                      LD      (HL),BLANK      ;ASSUME IT'S NOT THIS ONE
0120   2002                      JR      NZ,RD2          ;IF IT'S NOT
0122   365D                      LD      (HL),RARR       ;ELSE DISPLAY ARROW
0124   23                RD2:    INC     HL              ;LEAVE A SPACE
0125   37                        SCF
0126   CD7401                    CALL    RNAM            ;OUTPUT REG NAME
0129   CD7401                    CALL    RNAM
012C   3E04                      LD      A,4             ;MIGHT IT BE THE ALT SET?
012E   B8                        CP      B
012F   CCFA04                    CALL    Z,CLSOPN        ;###PAUSE IF B = 4
0132   380A                      JR      C,RD3           ;IF PC/SP/IY/IX
0134   3A0A00                    LD      A,(RSET)        ;(RSET) = 0 FOR STANDARD SET
0137   B7                        OR      A
0138   3680                      LD      (HL),BLANK      ;ASSUME NOT
013A   2802                      JR      Z,RD3           ;IF STANDARD SET
013C   3627                      LD      (HL),''''       ;ELSE APPEND A '
013E   23                RD3:    INC     HL              ;LEAVE A GAP
013F   23                        INC     HL
0140   DD5601                    LD      D,(IX+1)        ;DE <- REG CONTENTS
0143   DD5E00                    LD      E,(IX+0)
0146   DD2B                      DEC     IX
0148   DD2B                      DEC     IX
014A   CD8F06                    CALL    .DEOUT          ;DISPLAY CONTENTS
014D   100C                      DJNZ    RD5             ;FOR ALL BUT AF, ALSO DISPLAY (REG)
014F   23                        INC     HL              ;AF, DISPLAY FLAGS
0150   7B                        LD      A,E             ;A <- FLAG REG
0151   0608                      LD      B,8
0153   17                RD4:    RLA                     ;CY <- NEXT FLAG
0154   CD7401                    CALL    RNAM            ;DISPLAY FLAG IF CY SET
0157   10FA                      DJNZ    RD4
0159   183D                      JR      MDIS
015B   E5                RD5:    PUSH    HL              ;SAVE VT ADDR
015C   21FCFF                    LD      HL,-4           ;DISPLAY (REG-4) TO (REG+3)
015F   19                        ADD     HL,DE
```

```
0160   EB               EX     DE,HL          ;(DE) WILL BE DISPLAYED
0161   E1               POP    HL
0162   48               LD     C,B            ;SAVE B
0163   0608             LD     B,8
0165   23        RD6:   INC    HL
0166   1A               LD     A,(DE)         ;OUTPUT (REG)
0167   13               INC    DE
0168   CD9406           CALL   .BYTEO
016B   10F8             DJNZ   RD6
016D   41               LD     B,C            ;RESTORE B
016E   111F00           LD     DE,40H-33      ;DE <- DISTANCE TO NEXT LINE ON VT
0171   19               ADD    HL,DE          ;UPDATE VT ADDR
0172   18A4             JR     RD1

0174   3680      RNAM:  LD     (HL),BLANK
0176   3004             JR     NC,RN1         ;SKIP IF NO CARRY
0178   FD4E00           LD     C,(IY+0)       ;ELSE OUTPUT NEXT CHAR FROM RNSTR
017B   71               LD     (HL),C
017C   23        RN1:   INC    HL
017D   FD23             INC    IY
017F   C9               RET

0180   50435350  RNSTR: DEFM   'PCSPIYIXHLDEBCAFSZ H VNC'

                 ;MDIS - DISPLAY MEMORY AROUND MPTR

0198   CDFA04    MDIS:  CALL   CLSOPN         ;###PAUSE TO AVOID BLINK
019B   2A0740           LD     HL,(MPTR)      ;DE <- (MPTR)-12
019E   11F4FF           LD     DE,-12
01A1   19               ADD    HL,DE
01A2   EB               EX     DE,HL
01A3   21800A           LD     HL,0A80H       ;HL <- START OF DISPLAY
01A6   0604             LD     B,4            ;4 COLUMNS
01A8   48        MD1:   LD     C,B            ;COUNTED BY C
01A9   79               LD     A,C            ;###IF C = 2
01AA   FE02             CP     2              ;###
01AC   CCFA04           CALL   Z,CLSOPN       ;###PAUSE TO AVOID BLINK
01AF   0608             LD     B,8            ;8 ROWS
01B1   CD8F06    MD2:   CALL   .DEOUT         ;OUTPUT ADDRESS
01B4   23               INC    HL             ;LEAVE A BLANK
01B5   1A               LD     A,(DE)         ;GET CONTENT
01B6   CD9406           CALL   .BYTEO         ;OUTPUT IT
01B9   13               INC    DE             ;POINT TO NEXT
01BA   D5               PUSH   DE
01BB   113900           LD     DE,57          ;BUMP VT POINTER
01BE   19               ADD    HL,DE
01BF   D1               POP    DE
01C0   10EF             DJNZ   MD2            ;REPEAT FOR 8 ROWS
01C2   D5               PUSH   DE
01C3   110BFE           LD     DE,-200H+11    ;START OF NEXT COLUMN
01C6   19               ADD    HL,DE
01C7   D1               POP    DE
01C8   41               LD     B,C
01C9   10DD             DJNZ   MD1            ;REPEAT FOR 4 COLUMNS
```

```
01CB   F70C                  EMT     CLOSE           ;###

                     ;CMD - GET MODIFIED VALUE FOR MPTR OR REG, PARSE COMMAND

01CD   21F300        CMD:    LD      HL,RDIS         ;RETURN ADDRESS
01D0   E5                    PUSH    HL
01D1   F713                  EMT     GETHEX
01D3   3E2E                  LD      A,'.'           ;IF TERMINATOR WAS '.'
01D5   90                    SUB     B               ;THIS IS A REG OPERATION
01D6   2830                  JR      Z,SETR
01D8   79                    LD      A,C             ;GET NUMBER OF HEX CHARS
01D9   B7                    OR      A
01DA   2809                  JR      Z,CMD1          ;IF THERE WERE NONE
01DC   7D                    LD      A,L             ;A <- BYTE
01DD   2A0740                LD      HL,(MPTR)
01E0   F70B                  EMT     OPNWT           ;###IN CASE (MPTR) IS VT
01E2   77                    LD      (HL),A          ;STORE IT
01E3   F70C                  EMT     CLOSE           ;###
01E5   78            CMD1:   LD      A,B             ;A <- TERMINATOR
01E6   213002                LD      HL,TBL1
01E9   1600                  LD      D,0
01EB   42                    LD      B,D
01EC   0E08                  LD      C,8             ;BC <- TBL LENGTH (1ST)
01EE   EF37                  CALR    LOOKUP
01F0   200C                  JR      NZ,CMD2         ;IF NO MATCH
01F2   2A0740                LD      HL,(MPTR)       ;UPDATE MPTR
01F5   01F0FF                LD      BC,-16          ;VALUES FROM TBL ARE OFFSET
01F8   09                    ADD     HL,BC           ; TO SAVE INSTRUCTIONS
01F9   19                    ADD     HL,DE           ;ADD UPDATE
01FA   220740        SVM:    LD      (MPTR),HL
01FD   C9                    RET

01FE   0E18          CMD2:   LD      C,TBL2LN        ;BC <- TBL LENGTH (2ND)
0200   EF25                  CALR    LOOKUP
0202   E0                    RET     PO              ;NOTHING FOUND
0203   215002                LD      HL,SETM         ;CALCULATE REL ADDR
0206   19                    ADD     HL,DE
0207   E9                    JP      (HL)

                     ;SETR - MODIFY REG OR MOVE REG POINTER

0208   57            SETR:   LD      D,A             ;CLEAR D FOR LATER (A = 0)
0209   B9                    CP      C               ;TEST IF ANY HEX
020A   3A0940                LD      A,(RPTR)        ;A <- OFFSET FROM IX TO REG
020D   280A                  JR      Z,SR1           ;IF NO HEX
020F   5F                    LD      E,A             ;ELSE LOAD REG
0210   DD19                  ADD     IX,DE           ;POINT TO REG
0212   DD7401                LD      (IX+1),H        ;REG <- HL
0215   DD7500                LD      (IX+0),L
0218   C9                    RET                     ;OR 'NOP'
0219   3D            SR1:    DEC     A               ;DECR OFFSET, TO NEXT REG
021A   3D                    DEC     A
021B   2002                  JR      NZ,SR2          ;UNLESS OFF THE BOTTOM
021D   3E10                  LD      A,16            ;POINT TO PC
```

```
021F  FE0E            SR2:    CP      14              ;SKIP SP
0221  28F6                    JR      Z,SR1
0223  320940                  LD      (RPTR),A        ;SAVE OFFSET
0226  C9                      RET

                      ;LOOKUP - SEARCH TABLE POINTED TO BY HL FOR BC BYTES,
                      ;RETURNING ENTRY IN E
                      ;Z SET FOR A MATCH, P ODD IF OFF THE END

0227  EDA1            LOOKUP: CPI
0229  5E                      LD      E,(HL)
022A  C8                      RET     Z               ;IF MATCH
022B  EDA1                    CPI
022D  E0                      RET     PO              ;IF OFF THE END
022E  18F7                    JR      LOOKUP

                      ;TBL - FRONT PANEL SWITCHES

0230  0D              TBL1:   DEFB    CR              ;FORWARD ONE
0231  11                      DEFB    17
0232  2D                      DEFB    '-'             ;BACK ONE
0233  0F                      DEFB    15
0234  0A                      DEFB    LF              ;FORWARD EIGHT
0235  18                      DEFB    24
0236  2F                      DEFB    '/'             ;BACK EIGHT
0237  08                      DEFB    8
0238  47              TBL2:   DEFB    'G'             ;GET AND FIND PATTERN
0239  87                      DEFB    GETPAT-SETM
023A  48                      DEFB    'H'             ;HEX CALCULATOR
023B  6B                      DEFB    HEXCAL-SETM
023C  49                      DEFB    'I'             ;MPTR <- CURRENT WORD
023D  05                      DEFB    IND-SETM
023E  4A                      DEFB    'J'             ;CLEAR STACK, JUMP
023F  C3                      DEFB    FPJMP-SETM
0240  4B                      DEFB    'K'             ;CONTINUE EXECUTION
0241  39                      DEFB    CONTIN-SETM
0242  4D                      DEFB    'M'             ;SET MPTR
0243  00                      DEFB    SETM-SETM
0244  4E                      DEFB    'N'             ;FIND NEXT PATTERN
0245  9E                      DEFB    GPNXT-SETM
0246  50                      DEFB    'P'             ;FILL AND TEST MEMORY
0247  47                      DEFB    PUT-SETM
0248  52                      DEFB    'R'             ;MPTR <- MPTR + CURRENT OFFSET
0249  0E                      DEFB    REL-SETM
024A  53                      DEFB    'S'             ;MOVE MEMORY
024B  E5                      DEFB    MOVER-SETM
024C  57                      DEFB    'W'             ;SWITCH REGISTER SET
024D  20                      DEFB    SWITCH-SETM
024E  5A                      DEFB    'Z'             ;SINGLE STEP
024F  31                      DEFB    STEP-SETM
      0018            TBL2LN  EQU     $-TBL2

                      ;SETM - PROMPT AND GET NEW VALUE FOR MPTR
```

```
0250    CD0006        SETM:   CALL    PGTHEX
0253    18A5                  JR      SVM

                      ;IND - SET MPTR TO CURRENT WORD ADDRESS

0255    2A0740        IND:    LD      HL,(MPTR)
0258    5E                    LD      E,(HL)
0259    23                    INC     HL
025A    56                    LD      D,(HL)
025B    EB                    EX      DE,HL
025C    189C                  JR      SVM

                      ;REL - SET MPTR TO CURRENT RELATIVE ADDRESS

025E    2A0740        REL:    LD      HL,(MPTR)
0261    EF02                  CALR    RTOA            ;HL <- HL + (HL) + 1
0263    1895                  JR      SVM

                      ;RTOA - LOAD HL WITH HL + (HL) + 1 (REL TO ABS)
                      ;DESTROYS AF AND DE

0265    5E            RTOA:   LD      E,(HL)          ;E <- OFFSET
0266    1600                  LD      D,0             ;EXTEND SIGN TO D
0268    CB7B                  BIT     7,E
026A    2801                  JR      Z,RA1
026C    15                    DEC     D
026D    19            RA1:    ADD     HL,DE
026E    23                    INC     HL
026F    C9                    RET

                      ;SWITCH - SWITCH REGISTER SET BEING DISPLAYED

0270    3A0A40        SWITCH: LD      A,(RSET)        ;TOGGLE THE SWITCH
0273    2F                    CPL
0274    320A40                LD      (RSET),A
0277    F1                    POP     AF              ;POP RET ADDR
0278    F1                    POP     AF              ;RESTORE CURRENT SET
0279    C1                    POP     BC
027A    D1                    POP     DE
027B    E1                    POP     HL
027C    08                    EX      AF,AF'          ;EXCHANGE REGISTERS
027D    D9                    EXX
027E    C3E300                JP      FPM2            ;RE-ENTER TO DISPLAY OTHER SET

                      ;STEP - ENABLE SINGLE INSTRUCTION MODE
                      ;       MODE IS CLEARED ON NMI

0281    210340        STEP:   LD      HL,MASK         ;SET SS BIT IN MASK
0284    CB8E                  RES     SINGLE,(HL)
0286    23                    INC     HL              ;POINT TO SS FLG
0287    3653                  LD      (HL),'S'        ;SET MONITOR SS FLAG

                      ;CONTIN - CONTINUE EXECUTION AT SAVED PC
                      ;          CALL UPDATE TO ENABLE
```

```
                  ;          SINGLE STEP LOGIC IF SET IN 'MASK'

0289    F1        CONTIN: POP     AF               ;POP FP RET ADDR
028A    F1                POP     AF               ;POP CURRENT REGS
028B    C1                POP     BC
028C    D1                POP     DE
028D    E1                POP     HL
028E    DDE1              POP     IX
0290    F712              EMT     UPDATE           ;SS ON IF SET
0292    FDE1              POP     IY
0294    33                INC     SP
0295    33                INC     SP
0296    C9                RET
```

```
                    *E
                    ;PUT - FILLS AND TESTS MEMORY BETWEEN LIMITS

0297    CDE205      PUT:    CALL    FSTLST          ;DE <- FIRST, HL <- DIFF
029A    23                  INC     HL              ;LENGTH INCLUSIVE
029B    E5                  PUSH    HL              ;SAVE COUNT
029C    2134C0              LD      HL,MSGP
029F    CDF805              CALL    MGTHEX          ;GET FILL BYTE
02A2    7D                  LD      A,L             ;A = FILLER
02A3    EB                  EX      DE,HL           ;HL = START
02A4    C1                  POP     BC              ;BC = NO. OF ADDRS TO FILL

02A5    77          PT1:    LD      (HL),A          ;FILL MEMORY
02A6    EDA1                CPI                     ;THEN CHECK IT
02A8    280E                JR      Z,PT2           ;IF (HL) MATCHES A
02AA    F5                  PUSH    AF              ;ELSE ERR, SAVE PARITY FLAG
02AB    E5                  PUSH    HL              ;SAVE NEXT ADDR TO TEST
02AC    2B                  DEC     HL              ;(HL) IS NOW INCORRECT BYTE
02AD    220740              LD      (MPTR),HL       ;SO IT'S SEEN ON FP
02B0    CDC905              CALL    ERROUT          ;ERROR MSG
02B3    E1                  POP     HL              ;RESTORE NEXT ADDR
02B4    F1                  POP     AF              ;RESTORE PARITY FLAG
02B5    CDDE00              CALL    FPM1            ;FP ON, CONTINUE TEST AFTER 'K'
02B8    E0          PT2:    RET     PO              ;UNTIL BC = 0
02B9    18EA                JR      PT1
```

```
                    *E
                    ;HEXCAL - HEXADECIMAL CALCULATOR
02BB   CDFD05       HEXCAL: CALL    CGTHEX          ;GET X
02BE   E5                   PUSH    HL              ;(SP) <- X
02BF   EB                   EX      DE,HL           ;DE <- X
02C0   CDFD05               CALL    CGTHEX          ;GET Y
02C3   E5                   PUSH    HL              ;(SP) <- Y
02C4   19                   ADD     HL,DE           ;HL <- X+Y
02C5   EB                   EX      DE,HL           ;DE <- X+Y
02C6   21C60D               LD      HL,0DC6H        ;HL <- VT ADDR
02C9   CD8806               CALL    DEOUTW          ;DISPLAY SUM
02CC   D1                   POP     DE              ;GET Y
02CD   E3                   EX      (SP),HL         ;GET X, SAVE VT PTR
02CE   B7                   OR      A               ;CLEAR CY
02CF   ED52                 SBC     HL,DE           ;HL <- X-Y
02D1   EB                   EX      DE,HL           ;DE <- X-Y
02D2   E1                   POP     HL              ;RESTORE VT PTR
02D3   23                   INC     HL
02D4   C38806               JP      DEOUTW          ;DISP DIFF, RET TO FP
```

```
                    *E
                    ;GETPAT - ENTRY TO GET PATN FROM KBD, THEN SEARCH

02D7    114C40      GETPAT: LD      DE,PATN          ;DE <- PATN ADDR
02DA    AF                  XOR     A
02DB    324B40      GP1:    LD      (PATL),A         ;UPDATE PATN LENGTH
02DE    CDFD05              CALL    CGTHEX           ;POSSIBLY GET A BYTE
02E1    79                  LD      A,C              ;TEST FOR ONE
02E2    B7                  OR      A
02E3    2809                JR      Z,GPNXT          ;JUMP IF NO BYTE
02E5    7D                  LD      A,L              ;ELSE STORE IT
02E6    12                  LD      (DE),A
02E7    13                  INC     DE
02E8    3A4B40              LD      A,(PATL)         ;INC LENGTH
02EB    3C                  INC     A
02EC    18ED                JR      GP1

                    ;GPNXT - GET NEXT OCCURRENCE OF PATN

02EE                GPNXT:
02EE    2A0740      GP2:    LD      HL,(MPTR)        ;START FROM (MPTR)+1
02F1    23                  INC     HL
02F2    114C40              LD      DE,PATN          ;DE <- START OF PATTERN
02F5    1A                  LD      A,(DE)           ;A <- FIRST BYTE TO FIND
02F6    010000              LD      BC,0             ;SET BC FOR 65K
02F9    EDB1                CPIR                     ;FIND FIRST MATCH
02FB    2B                  DEC     HL               ;POINT TO START AND
02FC    220740              LD      (MPTR),HL        ;SAVE MEMORY PTR
02FF    23                  INC     HL               ;THEN POINT TO NEXT AGAIN
0300    3A4B40              LD      A,(PATL)         ;A <- PATTERN LENGTH
0303    B7                  OR      A                ;/IGNORE ZERO LENGTH
0304    C8                  RET     Z                ;/
0305    3D                  DEC     A                ;HAVE FOUND FIRST
0306    4F                  LD      C,A              ;BC <- LENGTH
0307    AF          GP3:    XOR     A
0308    47                  LD      B,A
0309    B1                  OR      C                ;WHILE (BC > 0)
030A    C8                  RET     Z
030B    13                  INC     DE               ;CHECK REST OF PATN
030C    1A                  LD      A,(DE)           ;A <- NEXT IN PATN
030D    EDA1                CPI
030F    20DD                JR      NZ,GP2           ;IF NO MATCH, BREAK TO CONTINUE SEARCH
0311    18F4                JR      GP3
```

```
                    *H FP/MONITOR ROUTINES

                    ;JUMP - PROMPT FOR ADDRESS, THEN
                    ;RESTORE SCROLLER AND GO TO ADDRESS

0313   310041       FPJMP:  LD      SP,STACK        ;CLEAN UP STACK

0316   CD0006       JUMP:   CALL    PGTHEX          ;GET ADDR
0319   E5                   PUSH    HL              ;SAVE FOR RET

                    ;SCROLL - RESTORES FULL SCREEN SCROLL

031A   010300       .SCROL: LD      BC,3            ;RESTORE SCROLLER

                    ;INIT - MOVES BC WORDS FROM INITBK TO RAM SCRATCH AREA

031D   21DE07       INIT:   LD      HL,INITBK
0320   110040       IT1:    LD      DE,RAM
0323   EDB0                 LDIR
0325   C9                   RET

                    ;GRAFIX - CLEARS GRAPHICS ('FRONT PANEL') AREA
                    ;AND SETS SCROLLER TO BOTTOM FOUR LINES

0326   3E14         .GRAFX: LD      A,20            ;NO. OF LINES
0328   210008               LD      HL,800H         ;TOP OF SCREEN
032B   F70F                 EMT     CLEAR

                    ;GRAFNC - SETS SCROLLER WITHOUT CLEARING

032D   010300       GRAFNC: LD      BC,3
0330   21FD07               LD      HL,GRAFBK
0333   18EB                 JR      IT1

                    ;MOVER - SEE MOVE

0335   186E         MOVER:  JR      MOVE
```

```
                  *H MONITOR ROUTINES
                  ;        'THE WHITE RABBIT PUT ON HIS SPECTACLES. 'WHERE
                  ;         SHALL I BEGIN, PLEASE YOUR MAJESTY?' HE ASKED.
                  ;         'BEGIN AT THE BEGINNING,' THE KING SAID GRAVELY,
                  ;         'AND GO ON TILL YOU COME TO THE END: THEN STOP.''
                  ;                 LEWIS CARROLL  ALICE'S ADVENTURES IN WONDERLAND
                  ;BEGIN - SET STACK AND SYSTEM POINTERS, DROP
                  ;THROUGH TO MAIN MONITOR COMMANDS
0337  310041      BEGIN:  LD      SP,STACK
033A  F3                  DI
033B  011F00              LD      BC,INITLN       ;COPY INITBK TO RAM
033E  EFDD                CALR    INIT
0340  3EFF                LD      A,0FFH          ;SET REMAINING VECTORS TO OFFH
0342  0621                LD      B,FFLEN
0344  211F40              LD      HL,OUT1
0347  77          BG1:    LD      (HL),A
0348  23                  INC     HL
0349  10FC                DJNZ    BG1
034B  36C9                LD      (HL),0C9H       ;SET MONX TO 'RET'
034D  3E93                LD      A,MKINIT!1      ;/CLEAR KBD (RESETS PORTO)
034F  32FC0F              LD      (PORTO),A
0352  F712                EMT     UPDATE
0354  3E0C                LD      A,FORM          ;CLEAR SCREEN
0356  F716                EMT     OUTC
0358  224340      BG2:    LD      (HIMEM),HL      ;FIND TOP OF RAM
035B  23                  INC     HL
035C  7E                  LD      A,(HL)          ;SAVE CONTENT
035D  2F                  CPL
035E  77                  LD      (HL),A          ;TEST IF COMPL READS BACK
035F  BE                  CP      (HL)
0360  2F                  CPL
0361  77                  LD      (HL),A          ;RESTORE ORIGINAL CONTENT
0362  28F4                JR      Z,BG2           ;CONTINUE UNTIL END OF RAM
0364  3A0010              LD      A,(NXTROM)      ;TEST FOR ADDITIONAL ROM
0367  B7                  OR      A               ;IF IT CONTAINS ZERO
0368  CC0010              CALL    Z,NXTROM        ; CALL IT
036B  21D607              LD      HL,MSGV         ;OUTPUT VERSION
036E  CDCC05              CALL    MOUT
                  ;CONTC - DO PARTIAL RESET, THEN DROP THROUGH TO MONITOR
0371  310041      CONTC:  LD      SP,STACK
0374  010400              LD      BC,CCLEN        ;RESET SCROLLER TO FULL SCREEN
0377  EFA4                CALR    INIT            ;AND INITIALISE PORTO MASK
                  ;ZMON - MONITOR COMMAND LOOP
```

```
0379  3E5D            ZMON:   LD      A,RARR          ;ISSUE CR, MONITOR PROMPT
037B  CD3A05                   CALL    AOUT
037E  CD4040                   CALL    MONX            ;CHECK FOR EXTENDED MONITOR
0381  CDCD00                   CALL    GETC
0384  F716                     EMT     OUTC
0386  FE4C                     CP      'L'             ;LOAD MEMORY FROM CASSETTE
0388  283F                     JR      Z,LOAD
038A  FE44                     CP      'D'             ;DUMP MEMORY TO CASSETTE
038C  CA6204                   JP      Z,SAVE          ;/POSSIBLY JR
038F  FE4A                     CP      'J'             ;GO TO ADDRESS
0391  2883                     JR      Z,JUMP
0393  FE43                     CP      'C'             ;REENTER USER PROG
0395  2807                     JR      Z,RNTR
0397  FE53                     CP      'S'             ;MOVE MEMORY
0399  CCA503                   CALL    Z,MOVE          ; MUST BE LAST IN LIST
039C  18DB                     JR      ZMON            ; BECAUSE MOVE DOES A 'RET'

                       ;RNTR - REENTER USER PROGRAM AT (ADDR) + 3

039E  2A0B40           RNTR:   LD      HL,(ADDR)
03A1  23                       INC     HL
03A2  23                       INC     HL
03A3  23                       INC     HL
03A4  E9                       JP      (HL)

                       ;MOVE - MOVE BLOCK OF MEMORY UP OR DOWN
                       ;       PROMPTS FOR START (OLD S) AND FINISH (OLD F),
                       ;        THEN NEW START (NEW S)

03A5  CDE205           MOVE:   CALL    FSTLST          ;DE <- OLD S, HL <- (OLD F-OLD S)
03A8  E5                       PUSH    HL              ;(SP) <- LEN-1
03A9  210600                   LD      HL,MSGM         ;GET NEW S
03AC  CDF805                   CALL    MGTHEX          ;HL <- NEW S
03AF  EB                       EX      DE,HL           ;HL <- OS, DE <- NS
03B0  C1                       POP     BC              ;BC <- LEN-1
03B1  E5                       PUSH    HL              ;(SP) <- OLD S
03B2  B7                       OR      A               ;CLEAR CY
03B3  ED52                     SBC     HL,DE           ;CY->UP, NO CY->DOWN
03B5  E1                       POP     HL              ;HL <- OLD S
03B6  F70B                     EMT     OPNWT           ;### ALLOW VT MOVES
03B8  3009                     JR      NC,MV1          ;IF MOVE DOWN
03BA  09                       ADD     HL,BC           ;HL <- OLD F
03BB  EB                       EX      DE,HL
03BC  09                       ADD     HL,BC
03BD  EB                       EX      DE,HL           ;DE <- NEW F
03BE  03                       INC     BC              ;MAKE LENGTH INCLUSIVE
03BF  EDB8                     LDDR                    ;MOVE UP FROM TOP
03C1  1803                     JR      MV2
03C3  03               MV1:    INC     BC              ;MAKE LENGTH INCLUSIVE
03C4  EDB0                     LDIR                    ;MOVE DOWN FROM BOTTOM
03C6  F70C             MV2:    EMT     CLOSE           ;###
03C8  C9                       RET

```

```
                    *H ABSOLUTE LOADER & DUMP

                    ;LOAD - LOADS TAPE IN RML ABSOLUTE FORMAT

03C9   CD5406       LOAD:   CALL    GETNAM          ;GET NAME TO FIND
03CC   CD4505               CALL    CROUT
                                                    ;START TAPE
03CF   210000       FIND:   LD      HL,0            ;/SET RECORD ADDR TO ZERO
03D2   CD4C04               CALL    RECNO           ;/
03D5   CD2904               CALL    GETHDR
03D8   3C                   INC     A               ;SET Z IF NAME REC (TYPE -1)
03D9   20F4                 JR      NZ,FIND         ;ELSE CONTINUE SEARCH

03DB   214C40               LD      HL,PATN         ;HL <- BUFFER ADDR FOR NAME
03DE   E5                   PUSH    HL              ;SAVE THE ADDR
03DF   CD3B04               CALL    GET             ;READ THE NAME REC
03E2   2025                 JR      NZ,LDERR        ;CKSUM ERROR
03E4   36FF                 LD      (HL),-1         ;PUT EOS TO BUF
03E6   E1                   POP     HL              ;HL <- START OF NAME
03E7   CDCF05               CALL    .MSG            ;DISPLAY IT (HL UNCHANGED)
03EA   CD4505               CALL    CROUT
03ED   114540               LD      DE,NAME         ;DE <- START OF WANTED NAME
03F0   0606                 LD      B,6             ;COMPARE 6 CHARS
03F2   1A           COMP:   LD      A,(DE)          ;GET CHAR FROM TARGET
03F3   BE                   CP      (HL)            ;COMPARE CHAR FROM THIS NAME
03F4   20D9                 JR      NZ,FIND         ;JUMP IF NO MATCH
03F6   13                   INC     DE              ;INC POINTERS
03F7   23                   INC     HL
03F8   10F8                 DJNZ    COMP            ;REPEAT IF MORE

03FA   CD2904       GETREC: CALL    GETHDR
03FD   E5                   PUSH    HL              ;SAVE ADDR
03FE   B7                   OR      A               ;TEST TYPE
03FF   2017                 JR      NZ,GR1          ;IF NOT TYPE 0
0401   CD3B04               CALL    GET             ;LOAD DATA
0404   E1                   POP     HL              ;/RESTORE ADDR
0405   EF45                 CALR    RECNO           ;/AND DISPLAY IT
0407   28F1                 JR      Z,GETREC        ;Z MEANS CKSUM OK

0409   CDQ905       LDERR:  CALL    ERROUT          ;OUTPUT ERROR MSG
040C   CDCD00       LDER1:  CALL    GETC            ;WAIT FOR KBD
040F   FE4C                 CP      'L'
0411   20F9                 JR      NZ,LDER1        ;WAIT FOR 'L'
0413   CD4505               CALL    CROUT
0416   18E2                 JR      GETREC          ;CONTINUE LOADING

0418   3D           GR1:    DEC     A               ;TEST TYPE
0419   200B                 JR      NZ,GR2          ;IF NOT TYPE 1
041B   EF28                 CALR    TI              ;EOF, GET CKSUM
041D   E1                   POP     HL              ;/RESTORE ADDR
041E   EF2C                 CALR    RECNO           ;/AND DISPLAY IT
0420   20E7                 JR      NZ,LDERR
                                                    ;STOP TAPE
0422   220B/0               LD      (ADDR),HL       ;SAVE START ADDR
```

3-23.18

```
0425   E9                        JP      (HL)             ;GO TO ADDRESS
0426   E1              GR2:      POP     HL               ;/SKIP REC IF NOT TYPES 0 OR 1
0427   18D1                      JR      GETREC           ;CONTINUE LOADING
                       ;GETHDR - GET RECORD HEADER
                       ; RETURN WITH HL=ADDRESS, B=LENGTH AND A=TYPE
0429   3E4D            GETHDR:   LD      A,SOH            ;WAIT FOR START OF RECORD
042B   F711                      EMT     GETSYN
042D   AF                        XOR     A                ;CLEAR CKSUM
042E   4F                        LD      C,A
042F   EF14                      CALR    TI               ;GET LENGTH
0431   47                        LD      B,A
0432   EF11                      CALR    TI               ;GET ADDR H
0434   67                        LD      H,A
0435   EF0E                      CALR    TI               ;GET ADDR L
0437   6F                        LD      L,A
0438   EF0B                      CALR    TI               ;GET RECORD TYPE
043A   C9                        RET
                       ;GET - READ B BYTES FROM TAPE TO (HL), RETURNING
                       ;         WITH Z SET FROM CKSUM
043B   EF08            GET:      CALR    TI
043D   77                        LD      (HL),A
043E   BE                        CP      (HL)
043F   2801                      JR      Z,GT1            ;IF BYTE LOADED
0441   FF                        RST     38H              ;ELSE BREAK
0442   23              GT1:      INC     HL
0443   10F6                      DJNZ    GET              ;DROP THRO' TO TI FOR CKSUM
                       ;TI - READ BYTE FROM TAPE INTO A, FORM CKSUM
0445   F704            TI:       EMT     GETBYT
0447   57                        LD      D,A              ;D <- BYTE
0448   81                        ADD     A,C              ;ADD CKSUM TO BYTE
0449   4F                        LD      C,A              ;RESTORE CKSUM
044A   7A                        LD      A,D              ;RESTORE BYTE
044B   C9                        RET                      ;WITH Z FLAG FROM CKSUM
                       ;RECNO - INDICATE TAPE ACTIVITY, DISPLAY RECORD ADDRESS
044C   E5              RECNO:    PUSH    HL               ;STANDARD SAVE SEQUENCE
044D   D5                        PUSH    DE
044E   C5                        PUSH    BC
044F   F5                        PUSH    AF
0450   EB                        EX      DE,HL            ;/DE <- ADDR (HL ON ENTRY)
0451   21C00D                    LD      HL,0DC0H         ;HL <- CURSOR ADDR
0454   F70B                      EMT     OPNWT            ;###
0456   7E                        LD      A,(HL)           ;BLINK CURSOR
0457   EE7F                      XOR     CURSOR
0459   77                        LD      (HL),A
045A   23                        INC     HL               ;/LEAVE A SPACE
```

```
045B    23          0883          INC     HL                  ;/
045C    CD8A06      0884          CALL    DEDUTC              ;/##} AND CLOSE
045F    C3C405      0885          JP      RET1                ;STANDARD EXIT
                    0886
```

```
                *E
                ;SAVE - DUMPS MEMORY TO TAPE WITH OPTIONAL START ADDR
0462  CD5406    SAVE:   CALL    GETNAM          ;GET A NAME
0465  CDE205            CALL    FSTLST          ;DE <- FIRST ADDR, BC <- LAST
0468  C5                PUSH    BC              ;(SP) <- LAST
0469  212300            LD      HL,MSGS         ;GET AUTOSTART ADDR
046C  CDF805            CALL    MGTHEX          ;HL <- START
046F  E3                EX      (SP),HL         ;(SP) <- START ADDR
0470  E5                PUSH    HL              ;SAVE LAST ADDR
0471  D5                PUSH    DE              ;SAVE FIRST ADDR
0472  CD4505            CALL    CROUT
0475  210000            LD      HL,0            ;/CLEAR REC ADDR
0478  EFD2              CALR    RECNO           ;/INDICATE ACTIVITY
                                                ;START TAPE
047A  CD9607            CALL    SEC5            ;5 SEC DELAY
047D  214540            LD      HL,NAME         ;OUTPUT NAME
0480  16FF              LD      D,-1            ;REC TYPE -1 (NAME)
0482  0606              LD      B,6
0484  CDB404            CALL    PUTRNA          ;PUTREC, NO ADDR
0487  D1                POP     DE              ;RESTORE DUMP LIMITS
0488  E1                POP     HL
0489  E5        SV1:    PUSH    HL              ;(SP) <- LAST ADDR
048A  067A              LD      B,RECLEN-6      ;B <- RECORD SIZE
048C  AF                XOR     A               ;CLEAR CY
048D  ED52              SBC     HL,DE           ;GET LENGTH TO DUMP
048F  3813              JR      C,SV3           ;DUMP COMPLETE (BECAUSE INCL)
0491  BC                CP      H               ;IF LENGTH > 255
0492  2006              JR      NZ,SV2          ; CONTINUE
0494  7D                LD      A,L             ;IF LENGTH >= 1 REC
0495  B8                CP      B
0496  3002              JR      NC,SV2          ; CONTINUE
0498  47                LD      B,A             ;ELSE DUMP REMAINDER
0499  04                INC     B               ; INCLUSIVELY
049A  EB        SV2:    EX      DE,HL           ;NOW HL <- CURRENT ADDR
049B  1600              LD      D,0             ;REC TYPE 0 (DATA)
049D  CDB204            CALL    PUTREC          ;WHICH UPDATES HL
04A0  D1                POP     DE              ;DE <- LAST ADDR TO DUMP
04A1  EB                EX      DE,HL           ;HL <- LAST, DE <- CURRENT
04A2  18E5              JR      SV1
04A4  E1        SV3:    POP     HL              ;END REC, BALANCE STACK
04A5  E1                POP     HL              ;HL <- START ADDR
04A6  0600              LD      B,0
04A8  1601              LD      D,1             ;REC TYPE 1 (EOF)
04AA  CDB204            CALL    PUTREC
04AD  CD9607            CALL    SEC5            ;5 SEC DELAY
                                                ;STOP TAPE
04B0  F700              EMT     0               ;BACK TO MONITOR
                ;PUTREC - WRITE RECORD TO TAPE WITH CKSUM
                ; OUTPUT B BYTES FROM (HL), D=TYPE, C=CKSUM
```

```
04B2    EF98            PUTREC: CALR    RECNO           ;/DISPLAY ADDR (FROM HL)
04B4    3E4D            PUTRNA: LD      A,SOH           ;PUT SOH (NOT IN CKSUM)
04B6    EF23                    CALR    TO
04B8    AF                      XOR     A               ;ZERO CKSUM
04B9    4F                      LD      C,A
04BA    78                      LD      A,B             ;PUT LENGTH
04BB    EF1E                    CALR    TO
04BD    7C                      LD      A,H             ;PUT ADDR H
04BE    EF1B                    CALR    TO
04C0    7D                      LD      A,L             ;PUT ADDR L
04C1    EF18                    CALR    TO
04C3    7A                      LD      A,D             ;PUT TYPE
04C4    EF15                    CALR    TO
04C6    78                      LD      A,B             ;CHECK LENGTH
04C7    B7                      OR      A
04C8    2806                    JR      Z,PR2           ;JUMP IF NO DATA
04CA    7E              PR1:    LD      A,(HL)
04CB    23                      INC     HL
04CC    EF0D                    CALR    TO
04CE    10FA                    DJNZ    PR1
04D0    79              PR2:    LD      A,C             ;GET CKSUM
04D1    ED44                    NEG
04D3    EF06                    CALR    TO              ;PUT CKSUM
04D5    0602                    LD      B,GAP
04D7    CD9807                  CALL    MS100           ;INTER RECORD GAP
04DA    C9                      RET

                        ;TO - PUT BYTE TO TAPE, FORM CKSUM IN C

04DB    F703            TO:     EMT     PUTBYT          ;A <- BYTE
04DD    81                      ADD     A,C             ;ADD CKSUM TO BYTE
04DE    4F                      LD      C,A             ;SAVE CKSUM
04DF    C9                      RET
```

```
                      *H GENERAL I/O ROUTINES

                      ;.KBDIN - READ KEYBOARD (CALLED VIA EMT)
                      ;ON EXIT Z SET IF NO CHAR, ELSE CLEAR
                      ;WATCH FOR CTRL C (RETURN TO MONITOR)

04E0    E5            .KBDIN: PUSH    HL
04E1    210340                LD      HL,MASK
04E4    CBC6                  SET     CLRKBD,(HL)     ;SET UP TO CLEAR KBD
04E6    3AFC0F                LD      A,(KBD)         ;A <-- POSSIBLE CHAR
04E9    E67F                  AND     7FH             ;CLEAR MS BIT
04EB    2802                  JR      Z,KBD1          ;IF NO CHAR AVAILABLE
04ED    F712                  EMT     UPDATE          ;CLEAR KBD
04EF    CB86          KBD1:   RES     CLRKBD,(HL)
04F1    F712                  EMT     UPDATE
04F3    E1                    POP     HL
04F4    C8                    RET     Z               ;NO CHAR AVAILABLE
04F5    FE03                  CP      CTRLC           ;IF NOT CTRL C
04F7    C0                    RET     NZ              ;RETURN THE CHAR
04F8    F700                  EMT     0               ;ELSE RESTART MONITOR

                      ;CLSOPN - CLOSE THEN OPEN (USED WHEN ROUTINE IS
                      ;LONGER THAN FRAME BLANKING PERIOD)

04FA    F70C          CLSOPN: EMT     CLOSE           ;DROP THRO' TO OPNWT


                      ;.OPNWT - OPEN VT MEMORY IN FRAME BLANKING PERIOD

04FC    E5            .OPNWT: PUSH    HL
04FD    F5                    PUSH    AF
04FE    21FE0F                LD      HL,PORT1
0501    CB76          OPNW1:  BIT     FRAME,(HL)      ;IF IN IT, WAIT
0503    20FC                  JR      NZ,OPNW1
0505    CB76          OPNW2:  BIT     FRAME,(HL)      ;WAIT FOR NEXT
0507    28FC                  JR      Z,OPNW2
0509    210340                LD      HL,MASK         ;OPEN MEMORY
050C    CBD6                  SET     VTOPN,(HL)
050E    F712                  EMT     UPDATE
0510    F1                    POP     AF
0511    E1                    POP     HL
0512    C9                    RET

                      ;.CLEAR - EXTERNAL CALL TO CLR WHICH DROPS
                      ;THROUGH TO CLOSE

0513    F70B          .CLEAR: EMT     OPNWT
0515    CD2B05                CALL    CLR

                      ;.CLOSE - CLOSE VT MEMORY

0518    E5            .CLOSE: PUSH    HL
0519    210340                LD      HL,MASK
051C    CB96                  RES     VTOPN,(HL)
```

3-23.23

```
051E   F712             EMT     UPDATE
0520   E1               POP     HL
0521   C9               RET

                ;UPDATE - OUTPUT MASK TO PORT 0

0522   F5       .UPDAT: PUSH    AF
0523   3A0340   UPD1:   LD      A,(MASK)
0526   32FC0F           LD      (PORTO),A
0529   F1               POP     AF
052A   C9               RET

                ;CLR - CLEAR THE VT SCREEN
                ;ON ENTRY A HOLDS THE NUMBER OF LINES TO
                ;CLEAR AND HL THE ADDRESS OF THE FIRST LINE
                ;ON EXIT HL POINTS TO ONE PAST LAST CHARACTER

052B   111800   CLR:    LD      DE,018H         ;INCR BETWEEN LINES
052E   0628     CLR1:   LD      B,40            ;CHARS/LINE
0530   3680     CLR2:   LD      (HL),BLANK      ;CLEAR A LINE
0532   23               INC     HL
0533   10FB             DJNZ    CLR2
0535   3D               DEC     A               ;ALL LINES DONE
0536   C8               RET     Z               ;YES
0537   19               ADD     HL,DE           ;POINT TO NEXT LINE
0538   18F4             JR      CLR1

                ;AOUT, PMTOUT AND CROUT OUTPUT TO THE VT SCREEN

053A   F5       AOUT:   PUSH    AF              ;OUTPUT CR THEN A
053B   CD4505           CALL    CROUT
053E   F1               POP     AF
053F   1806             JR      ..OUTC

0541   3E3E     PMTOUT: LD      A,'>'           ;OUTPUT PROMPT
0543   1802             JR      ..OUTC          ;RETURNING VIA OUTC

0545   3E0D     CROUT:  LD      A,CR            ;RETURN VIA OUTC
0547            ..OUTC:

                ;.OUTC - PUTS ONE CHARACTER TO SCREEN
                ;CALLED VIA 'EMT OUTC' BY MONITOR
                ;INTERPRETS DELETE, CR, FORM FEED AND TAB

0547   E5       .OUTC:  PUSH    HL              ;SAVE ALL
0548   D5               PUSH    DE
0549   C5               PUSH    BC
054A   F5               PUSH    AF
054B   F70B             EMT     OPNWT           ;###
054D   2A0540           LD      HL,(VTPTR)      ;HL <- CURRENT POS
0550   3680             LD      (HL),BLANK      ;CURSOR OFF
0552   118A05           LD      DE,SCRL         ;ELSE RET VIA SCRL
0555   D5               PUSH    DE
```

```
0556   FE7F                CP     DELETE         ;IF DELETE
0558   2006                JR     NZ,OUTC1
055A   3EC0                LD     A,0C0H            ;ANY CHARS?
055C   BD                  CP     L
055D   D0                  RET    NC                ;NO--IGNORE
055E   2B                  DEC    HL                ;YES--DEC PTR
055F   C9                  RET

0560   FE20       OUTC1:   CP     20H            ;ELSE IF NOT CONTR CHAR
0562   3808                JR     C,OUTC2
0564   FE4F                CP     'O'               ;SWITCH 'O'
0566   2001                JR     NZ,OUTC1A
0568   AF                  XOR    A
0569   77         OUTC1A:  LD     (HL),A            ;DISPLAY IT
056A   23                  INC    HL                ;AND INC PTR
056B   C9                  RET

056C   D60D       OUTC2:   SUB    CR             ;ELSE IF CARR RET
056E   2003                JR     NZ,OUTC3
0570   2EE8                LD     L,0E8H            ;FORCE A SCROLL
0572   C9                  RET

0573   3C         OUTC3:   INC    A              ;ELSE IF FORM FEED
0574   2008                JR     NZ,OUTC5
0576   3A0040              LD     A,(NLINES)        ;CLEAR SCROLLING SCREEN
0579   2A0140              LD     HL,(TOP)
057C   18AD                JR     CLR

                  ; OUTC4:  COULD BE VERTICAL TAB

057E   C603       OUTC5:   ADD    A,3            ;ELSE IF NOT HOR TAB
0580   C0                  RET    NZ                ;IGNORE
0581   7D                  LD     A,L            ;GET LAST TAB POS
0582   E6F8                AND    0F8H
0584   6F                  LD     L,A
0585   010800              LD     BC,8           ;BUMP BY 8
0588   09                  ADD    HL,BC
0589   C9                  RET

                  ;SCRL - SCROLL THE VT SCREEN IF NECESSARY, CLEARING
                  ;LAST LINE, THEN RESTORE CURSOR AND UPDATE POINTER

058A   3EE8       SCRL:    LD     A,0E8H
058C   BD                  CP     L              ;OFF THE END?
058D   202E                JR     NZ,CURON       ;NO

058F   3A0040              LD     A,(NLINES)
0592   3D                  DEC    A              ;A <- LINES TO SCROLL
0593   2A0140              LD     HL,(TOP)
0596   EB                  EX     DE,HL          ;DE <- ADDR OF TOP LINE
0597   214000              LD     HL,40H
059A   19                  ADD    HL,DE          ;HL <- ADDR OF NEXT LINE
059B   012800     SCRL1:   LD     BC,40          ;BC <- LINE LENGTH
059E   EDB0                LDIR
```

```
05A0    3D                      DEC     A
05A1    2810                    JR      Z,CLAST         ;DONE?
05A3    011800                  LD      BC,18H          ;NO
05A6    09                      ADD     HL,BC           ;ADVANCE POINTERS TO NEXT
05A7    EB                      EX      DE,HL
05A8    09                      ADD     HL,BC
05A9    EB                      EX      DE,HL
05AA    F5                      PUSH    AF
05AB    E607                    AND     7               ;###PAUSE EVERY 8
05AD    CCFA04                  CALL    Z,CLSOPN        ;###
05B0    F1                      POP     AF
05B1    18E8                    JR      SCRL1
05B3    3EC0            CLAST:  LD      A,0C0H          ;A <- START OF LAST
05B5    BD                      CP      L               ;DONE?
05B6    2805                    JR      Z,CURON         ;YES
05B8    2B                      DEC     HL
05B9    3680                    LD      (HL),BLANK
05BB    18F8                    JR      CLAST+2
05BD    367F            CURON:  LD      (HL),CURSOR
05BF    220540                  LD      (VTPTR),HL
05C2    F70C                    EMT     CLOSE           ;###
05C4    F1              RET1:   POP     AF              ;STANDARD EXITS
05C5    C1              RET2:   POP     BC
05C6    D1                      POP     DE
05C7    E1                      POP     HL
05C8    C9                      RET
                        ;MOUT - OUTPUT CR, THEN STRING, USING .MSG
                        ;  REG A LOST, OTHERS PRESERVED
05C9    211300          ERROUT: LD      HL,MSGE
05CC    CD4505          MOUT:   CALL    CROUT
                        ;.MSG - FAST STRING OUTPUT TO VT SCROLLER
                        ;  ENTER WITH (HL) -> STRING
                        ;  END OF STRING (EOS) IS A -VE BYTE
                        ;  ALL REGISTERS PRESERVED
05CF    E5              .MSG:   PUSH    HL              ;STANDARD SAVE SEQUENCE
05D0    D5                      PUSH    DE
05D1    C5                      PUSH    BC
05D2    F5                      PUSH    AF
05D3    ED5B0540                LD      DE,(VTPTR)
05D7    F70B                    EMT     OPNWT           ;###
05D9    EDA0            .MSG1:  LDI                     ;CAN ALWAYS MOVE ONE CHAR
05DB    CB7E                    BIT     7,(HL)          ;REACHED END OF STRING?
05DD    28FA                    JR      Z,.MSG1         ;CONTINUE IF NOT
05DF    EB                      EX      DE,HL           ;USE SCROLLER TO REPLACE
05E0    18A8                    JR      SCRL            ;CURSOR, DO CR IF NEEDED
                        ;FSTLST - PROMPT FOR 'FIRST' ADDR AND 'LAST' ADDR
```

```
                    ;ON RETURN, DE = FIRST, BC = LAST AND HL = (LAST - FIRST)
                    ;ERROR IF FIRST > LAST

05E2   211B00       FSTLST: LD      HL,MSGF         ;GET 'FIRST'
05E5   EF11                 CALR    MGTHEX
05E7   EB                   EX      DE,HL           ;DE <- FIRST
05E8   212B00               LD      HL,MSGL         ;GET 'LAST'
05EB   EF0B                 CALR    MGTHEX
05ED   44                   LD      B,H             ;BC <- LAST
05EE   4D                   LD      C,L
05EF   B7                   OR      A               ;TEST IF VALID, CLR CY
05F0   ED52                 SBC     HL,DE
05F2   D0                   RET     NC              ;WITH HL = LAST - FIRST

                    ;ERROR - ISSUE ERROR MESSAGE AND RESTART

05F3   CDC905       ERROR:  CALL    ERROUT
05F6   F700                 EMT     0               ;AND RESET STACK

                    ;GETHEX - GETS HEX VALUE FROM KEYBOARD
                    ;ON RETURN HL CONTAINS VALUE, C CONTAINS NUMBER
                    ;OF HEX CHARS, B CONTAINS TERMINATOR AND DE IS
                    ;UNCHANGED.  INTERPRETS DELETE.

05F8   CDCC05       MGTHEX: CALL    MOUT            ;OUTPUT CR, THEN STR >
05FB   1803                 JR      PGTHEX

05FD   CD4505       CGTHEX: CALL    CROUT           ;OUTPUT CR, >

0600   CD4105       PGTHEX: CALL    PMTOUT          ;OUTPUT PROMPT

0603   210000       .GTHEX: LD      HL,0            ;CLEAR BUFFER AND
0606   4D                   LD      C,L             ;CHAR COUNTER
0607   F702         NXTC:   EMT     KBDIN           ;GET A CHAR
0609   28FC                 JR      Z,NXTC
060B   47                   LD      B,A             ;SAVE LAST IN B
060C   FE30                 CP      '0'             ;CONTROL CHAR?
060E   D8                   RET     C               ;YES--RETURN
060F   FE7F                 CP      DELETE          ;IF DELETE
0611   281E                 JR      Z,NXTC2         ;DELETE LAST DIG (IF ANY)
0613   F716                 EMT     OUTC            ;ECHO PRINTING CHAR
0615   D630                 SUB     '0'             ;CONVERT TO HEX
0617   FE0A                 CP      10              ;0 TO 9?
0619   3808                 JR      C,NXTC1         ;YES
061B   D607                 SUB     7
061D   FE0A                 CP      0AH             ;LESS THAN A?
061F   D8                   RET     C               ;YES --RETURN
0620   FE10                 CP      10H             ;MORE THAN F?
0622   D0                   RET     NC              ;YES

0623   29           NXTC1:  ADD     HL,HL           ;SHIFT LEFT 4
0624   29                   ADD     HL,HL
0625   29                   ADD     HL,HL
0626   29                   ADD     HL,HL
```

```
0627  B5            OR      L              ;MERGE NEW
0628  6F            LD      L,A
0629  3E03          LD      A,3            ;HAVE THERE BEEN 4 CHARS?
062B  B9            CP      C
062C  3812          JR      C,NXTC4        ;IF C > 3
062E  0C            INC     C              ;ELSE BUMP CHAR COUNT
062F  18D6          JR      NXTC

0631  0C    NXTC2:  INC     C              ;DELETE, TEST IF ANY CHARS
0632  0D            DEC     C
0633  28D2          JR      Z,NXTC         ;Z IF NONE
0635  0604          LD      B,4            ;SHIFT RIGHT 4
0637  CB3C  NXTC3:  SRL     H
0639  CB1D          RR      L
063B  10FA          DJNZ    NXTC3
063D  0D            DEC     C              ;DEC COUNT
063E  180E          JR      NXTC5          ;ECHO DELETE

0640  D5    NXTC4:  PUSH    DE             ;5 CHARS, SAVE DE
0641  EB            EX      DE,HL          ;DE <- HEX VALUE
0642  2A0540        LD      HL,(VTPTR)     ;HL <- ADDR OF CURSOR
0645  7D            LD      A,L
0646  D605          SUB     5
0648  6F            LD      L,A            ;HL <- ADDR OF START OF STR
0649  CD8806        CALL    DEOUTW         ;DISPLAY TRUE 4 CHAR STR
064C  EB            EX      DE,HL          ;HL <- HEX VALUE
064D  D1            POP     DE             ;RESTORE DE

064E  3E7F  NXTC5:  LD      A,DELETE       ;DELETE FIFTH CHAR
0650  F716          EMT     OUTC
0652  18B3          JR      NXTC

            ;GETNAM - GET NAME FROM KBD, HANDLE DELETE

0654  21D107 GETNAM: LD     HL,MSGN        ;OUTPUT CR NAME>
0657  CDCC05        CALL    MOUT
065A  CD4105        CALL    FMTOUT
065D  0606          LD      B,6
065F  214540        LD      HL,NAME        ;HL <- BUFFER ADDR
0662  CDCD00 NM1:   CALL    GETC           ;GET A CHAR
0665  FE30          CP      '0'
0667  380A          JR      C,NM2
0669  FE7F          CP      DELETE
066B  280E          JR      Z,NM4
066D  F716          EMT     OUTC           ;ECHO CHAR
066F  77            LD      (HL),A         ;SAVE IT IN BUF
0670  23            INC     HL
0671  10EF          DJNZ    NM1            ;GET UP TO 6 CHARS
0673  04    NM2:    INC     B              ;WHILE (B != 0)
0674  05    NM3:    DEC     B
0675  C8            RET     Z
0676  3620          LD      (HL),' '       ;PAD WITH SPACE (NOT GRAPHICS BLANK)
0678  23            INC     HL
0679  18F9          JR      NM3
```

```
067B    78         NM4:   LD     A,B          ;TEST IF ANY CHARS
067C    FE06              CP     6
067E    28E2              JR     Z,NM1        ;Z IF NONE
0680    04                INC    B            ;ELSE UPDATE CNTR
0681    2B                DEC    HL           ;AND DELETE CHAR FROM BUF
0682    3E7F              LD     A,DELETE     ;ECHO DELETE
0684    F716              EMT    OUTC
0686    18DA              JR     NM1
```

```
                 *H VT OUTPUT ROUTINES
                 ;HL IS SCREEN ADDRESS

                 ;DEOUTW - DEOUT WHEN VT IS CLOSED
                 ;DEOUTC - DEOUT, THEN CLOSE

0688    F70B     DEOUTW: EMT     OPNWT           ;###
068A    F714     DEOUTC: EMT     DEOUT
068C    F70C             EMT     CLOSE           ;###
068E    C9               RET

                 ;.DEOUT - PUTS 4 HEX CHARS FROM DE

068F    7A       .DEOUT: LD      A,D
0690    CD9406           CALL    .BYTEO
0693    7B               LD      A,E

                 ;.BYTEO - PUTS 2 HEX CHARS FROM A

0694    F5       .BYTEO: PUSH    AF
0695    0F               RRCA                    ;GET UPPER 4 BITS
0696    0F               RRCA
0697    0F               RRCA
0698    0F               RRCA
0699    CD9D06           CALL    HEXO
069C    F1               POP     AF

                 ;HEXO - PUTS LOWER 4 BITS OF A IN HEX

069D    E60F     HEXO:   AND     0FH             ;MASK TO 4 BITS
069F    C690             ADD     A,90H           ;CONVERT TO HEX
06A1    27               DAA
06A2    CE40             ADC     A,40H
06A4    27               DAA
06A5    77       CHO:    LD      (HL),A          ;PUT OUT CHARACTER
06A6    23               INC     HL              ;BUMP POINTER
06A7    C9               RET
```

3-23.30

```
                  *H TAPE I/O ROUTINES

                  ;.GTBYT - GETS NEXT BYTE FROM TAPE INTO A
                  ;CALLED VIA EMT

06A8    E5        .GTBYT: PUSH    HL              ;SAVE ALL BUT A
06A9    D5                PUSH    DE
06AA    C5                PUSH    BC
06AB    CDF606            CALL    GETCYC
06AE    AF                XOR     A
06AF    57                LD      D,A
06B0    3D                DEC     A
06B1    5F                LD      E,A

06B2    CB5B      GB1:    BIT     3,E
06B4    2001              JR      NZ,GB2
06B6    15                DEC     D
06B7    CDF606    GB2:    CALL    GETCYC
06BA    3801              JR      C,GB3
06BC    14                INC     D
06BD    CB13      GB3:    RL      E
06BF    3E01              LD      A,1
06C1    BA                CP      D
06C2    30EE              JR      NC,GB1

06C4    CB5B              BIT     3,E
06C6    280C              JR      Z,GB4
06C8    CDF606            CALL    GETCYC
06CB    CB13              RL      E
06CD    CB5B              BIT     3,E
06CF    2803              JR      Z,GB4
06D1    CDF606            CALL    GETCYC
06D4    0608      GB4:    LD      B,8
06D6    50                LD      D,B

06D7    1E00      GB5:    LD      E,0
06D9    3EF8              LD      A,-8
06DB    82                ADD     A,D
06DC    57                LD      D,A

06DD    CDF606    GB6:    CALL    GETCYC
06E0    14                INC     D
06E1    1C                INC     E
06E2    3802              JR      C,GB7
06E4    14                INC     D
06E5    1D                DEC     E
06E6    3E07      GB7:    LD      A,7
06E8    BA                CP      D
06E9    30F2              JR      NC,GB6

06EB    3E03              LD      A,3
06ED    BB                CP      E
06EE    CB19              RR      C
06F0    10E5              DJNZ    GB5
```

```
06F2  79                     LD      A,C
06F3  C3C505                 JP      RET2            ;STANDARD EXIT

                    ;GETCYC - TIME A CYCLE FROM TAPE
                    ;/ALLOW ESCAPE BY CTRL C

06F6  21FE0F        GETCYC: LD      HL,PORT1
06F9  CDE004        GC1:    CALL    .KBDIN          ;CHECK FOR CTRL C
06FC  CB6E                  BIT     TAPE,(HL)
06FE  28F9                  JR      Z,GC1
0700  CB6E                  BIT     TAPE,(HL)
0702  28F5                  JR      Z,GC1
0704  CB6E          GC2:    BIT     TAPE,(HL)
0706  20FC                  JR      NZ,GC2
0708  CB6E                  BIT     TAPE,(HL)
070A  20F8                  JR      NZ,GC2
070C  CD7307                CALL    RDCNT
070F  CD8607                CALL    RESCNT
0712  FE47                  CP      VALUE
0714  C9                    RET

                    ;GETSYN - WAIT FOR START OF RECORD (TRAP ROUTINE)
                    ;CALLED WITH SOH IN A. WAITS FOR INTERRECORD GAP,
                    ;THEN SOH CHAR.  REGS UNCHANGED.

0715  E5            .GTSYN: PUSH    HL              ;STANDARD SAVE
0716  D5                    PUSH    DE
0717  C5                    PUSH    BC
0718  F5                    PUSH    AF
0719  4F                    LD      C,A             ;C <- SOH
071A  CD0D40        GTSN1:  CALL    GETGAP          ;WAIT FOR GAP
071D  F704                  EMT     GETBYT          ;READ FIRST CHAR
071F  B9                    CP      C               ;IS IT CORRECT?
0720  20F8                  JR      NZ,GTSN1        ;NO--ASSUME A GLITCH
0722  181F                  JR      PBY2            ;STANDARD EXIT

                    ;.GTGAP - FIND INTER RECORD GAP (GAPLEN '1' CYCLES)
                    ;CALLED VIA GETGAP (RAM VECTOR) SO THAT IT CAN BE
                    ;SKIPPED FOR PAPER TAPE

0724  06F0          .GTGAP: LD      B,GAPLEN-256    ;SET B TO # OF '1' CYCLES
0726  CDF606        GG1:    CALL    GETCYC
0729  30F9                  JR      NC,.GTGAP       ;RESET B IF A '0' CYCLE SEEN
072B  10F9                  DJNZ    GG1             ;COUNT DOWN B '1' CYCLES
072D  C9                    RET
```

3-23.32

```
                  *E
                  ;.PTBYT - OUTPUTS BYTE TO TAPE FROM A
                  ;CALLED VIA EMT

072E  E5          .PTBYT: PUSH   HL            ;SAVE ALL
072F  D5                  PUSH   DE
0730  C5                  PUSH   BC
0731  F5                  PUSH   AF
0732  2F                  CPL
0733  4F                  LD     C,A
0734  3E01                LD     A,1
0736  EF0E                CALR   PUTBIT
0738  060A                LD     B,10

073A  CB39        PBY1:   SRL    C
073C  17                  RLA
073D  EF07                CALR   PUTBIT
073F  10F9                DJNZ   PBY1

0741  F702                EMT    KBDIN
0743  C3C405      PBY2:   JP     RET1          ;STANDARD EXIT

                  ;PUTBIT - PUT BIT 0 OF A TO TAPE

0746  C5          PUTBIT: PUSH   BC
0747  210340              LD     HL,MASK
074A  0604                LD     B,4
074C  E603                AND    11B
074E  281F                JR     Z,PBT4
0750  FE03                CP     11B
0752  281B                JR     Z,PBT4
0754  CBA6                RES    FREQ,(HL)
0756  FE01                CP     01B
0758  2802                JR     Z,PBT1
075A  CBE6                SET    FREQ,(HL)

075C  F5          PBT1:   PUSH   AF
075D  7E                  LD     A,(HL)
075E  11FC0F              LD     DE,PORT0
0761  21FE0F              LD     HL,PORT1
0764  CB4E        PBT2:   BIT    CLK,(HL)
0766  28FC                JR     Z,PBT2
0768  CB4E        PBT3:   BIT    CLK,(HL)
076A  20FC                JR     NZ,PBT3
076C  12                  LD     (DE),A
076D  F1                  POP    AF
076E  05                  DEC    B

076F  F710        PBT4:   EMT    EDGE
0771  C1                  POP    BC
0772  C9                  RET
```

```
                  *E
                  ;RDCNT - READ THE 8 USEC COUNTER INTO A (REGS PRESERVED)

0773   E5         RDCNT:  PUSH    HL
0774   C5                 PUSH    BC
0775   21FD0F             LD      HL,CNTR
0778   46                 LD      B,(HL)
0779   7E                 LD      A,(HL)
077A   4E                 LD      C,(HL)
077B   66                 LD      H,(HL)
077C   B8                 CP      B
077D   2804               JR      Z,RC1
077F   B9                 CP      C
0780   2801               JR      Z,RC1
0782   7C                 LD      A,H
0783   C1         RC1:    POP     BC
0784   E1                 POP     HL
0785   C9                 RET

                  ;RESCNT - CLEAR THE 8 USEC COUNTER (REGS PRESERVED)
                  ;(N.B. TO RETAIN TIMING, DOES NOT USE TRAP FOR 'UPDATE')

0786   E5         RESCNT: PUSH    HL
0787   210340             LD      HL,MASK
078A   CBF6               SET     CLRCNT,(HL)
078C   CD2205             CALL    .UPDAT
078F   CBB6               RES     CLRCNT,(HL)
0791   CD2205             CALL    .UPDAT
0794   E1                 POP     HL
0795   C9                 RET

                  ;SEC5 - WAIT 5 SEC (DESTROYS B)

0796   0632       SEC5:   LD      B,50            ;SET 50 * 100 MSEC

                  ;MS100 - WAIT FOR B 100 MSEC INTERVALS

0798   C5         MS100:  PUSH    BC
0799   0678               LD      B,120           ;SET 120 * 1/1200 SEC
079B   F710               EMT     EDGE
079D   C1                 POP     BC
079E   10F8               DJNZ    MS100
07A0   C9                 RET

                  ;EDGE - WAIT FOR B 1200 HZ EDGES
                  ;       (ALSO USED AS 833 1/3 USEC TIMER)

07A1   E5         .EDGE:  PUSH    HL
07A2   D5                 PUSH    DE
07A3   C5                 PUSH    BC
07A4   F5                 PUSH    AF
07A5   21FE0F             LD      HL,PORT1
07A8   CB4E       ED1:    BIT     CLK,(HL)        ;WAIT FOR CLOCK HIGH
07AA   28FC               JR      Z,ED1
```

```
07AC    CB4E     ED2:   BIT     CLK,(HL)        ;WAIT FOR CLOCK LOW
07AE    20FC            JR      NZ,ED2
07B0    10F6            DJNZ    ED1
07B2    C3C405          JP      RET1

```

```
                *H ADDRESSES AND PARAMETERS

                ;DISPATCH TABLE FOR TRAP CODES 11 TO 24

      000B      OPNWT   EQU     11
      000C      CLOSE   EQU     12
      000D      GRAFIX  EQU     13
      000E      SCROLL  EQU     14
      000F      CLEAR   EQU     15
      0010      EDGE    EQU     16
      0011      GETSYN  EQU     17
      0012      UPDATE  EQU     18
      0013      GETHEX  EQU     19
      0014      DEOUT   EQU     20
      0015      BYTEO   EQU     21
      0016      OUTC    EQU     22
      0017      MSG     EQU     23
      0018      CHAN    EQU     24

07B5  FC04      TRTBL:  DEFW    .OPNWT
07B7  1805              DEFW    .CLOSE
07B9  2603              DEFW    .GRAFX
07BB  1A03              DEFW    .SCROL
07BD  1305              DEFW    .CLEAR
07BF  A107              DEFW    .EDGE
07C1  1507              DEFW    .GTSYN
07C3  2205              DEFW    .UPDAT
07C5  0306              DEFW    .GTHEX
07C7  8F06              DEFW    .DEOUT
07C9  9406              DEFW    .BYTEO
07CB  4705              DEFW    .OUTC
07CD  CF05              DEFW    .MSG
07CF  6E00              DEFW    .CHAN

                ;SYSTEM MESSAGES - THE MAJORITY OF THESE ARE INTERSPERSED
                ;WITH THE JUMPS TO THE 'RST' VECTORS

07D1  4E414D45  MSGN:   DEFM    'NAME'
07D5  FF                DEFB    -1
07D6  434F5320  MSGV:   DEFM    'COS 2.3'       ;VERSION NUMBER
07DD  FF                DEFB    -1

                ;INITBK - THIS THE INITIAL STATE OF RAM AND
                ;IS SET UP BY BEGIN ON RESET OR PARTIALLY BY CTRL C

07DE  18        INITBK: DEFB    24              ;NLINES
07DF  0008              DEFW    0800H           ;TOP
07E1  92                DEFB    MKINIT          ;MASK
      0004      CCLEN   EQU     $-INITBK
07E2  00                DEFB    0               ;SSFLG
07E3  C00D              DEFW    0DC0H           ;VTPTR
07E5  0041              DEFW    STACK           ;MPTR
07E7  10                DEFB    16              ;RPTR
07E8  00                DEFB    0               ;RSET
```

```
07E9    0000                      DEFW    0               ;ADDR
07EB    C32407                    JP      .GTGAP          ;GETGAP.
07EE    C34705                    JP      .OUTC
07F1    C3E004                    JP      .KBDIN          ;KBDIN
07F4    C32E07                    JP      .PTBYT          ;PUTBYT
07F7    C3A806                    JP      .GTBYT          ;GETBYT
07FA    C31040                    JP      VTV             ;LP VECTOR
        001F            INITLN    EQU     $-INITBK

07FD    04              GRAFBK:   DEFB    4               ;NLINES
07FE    000D                      DEFW    0D00H           ;TOP

        07FF            MONEND    EQU     $-1
        0800            MONLEN    EQU     MONEND-MONST+1

                        ;DEDICATED RAM FOR SYSTEM

0800                              ORG     RAM

4000                    NLINES:   DEFS    1               ;LINES TO SCROLL
4001                    TOP:      DEFS    2               ;TOP OF SCROLLED AREA
4003                    MASK:     DEFS    1               ;PORT 0 MASK
4004                    SSFLG:    DEFS    1               ;SINGLE STEP FLAG
4005                    VTPTR:    DEFS    2               ;VT POINTER
4007                    MPTR:     DEFS    2               ;MEMORY POINTER
4009                    RPTR:     DEFS    1               ;REGISTER POINTER
400A                    RSET:     DEFS    1               ;STANDARD/ALT REG SWITCH
400B                    ADDR:     DEFS    2               ;USER COLD START ADDR
400D                    GETGAP:   DEFS    3               ;GAP VECTOR (ALTER FOR PAPER TAPE)

                        ;TRANSFER VECTORS

        0002            KBDIN     EQU     2
        0003            PUTBYT    EQU     3
        0004            GETBYT    EQU     4

4010                    VTV:      DEFS    3               ;CONSOLE OUT
4013                    KBDV:     DEFS    3               ;CONSOLE IN
4016                    TOV:      DEFS    3               ;TAPE OUT
4019                    TIV:      DEFS    3               ;TAPE IN
401C                    LPV:      DEFS    3               ;LINE PRINTER OUT

                                                          ;HERE DOWN INITIALISED TO 'OFFH'
401F                    OUT1:     DEFS    3               ;AUX I/O, USED BY FILE SYSTEMS
4022                    OUT2:     DEFS    3
4025                    IN1:      DEFS    3
4028                    IN2:      DEFS    3
402B                    IN3:      DEFS    3

402E                    RST8:     DEFS    3               ;RESTART VECTORS
4031                    RST10:    DEFS    3
4034                    RST18:    DEFS    3
4037                    RST20:    DEFS    3

```

```
403A                 NMIX:   DEFS    3               ;NMI EXTENSION VECTOR
403D                 TRAPX:  DEFS    3               ;TRAP EXTENSION VECTOR
        0021         FFLEN   EQU     $-OUT1
4040                 MONX:   DEFS    3               ;MONITOR EXTENSION VECTOR
4043                 HIMEM:  DEFS    2               ;TOP OF RAM

                     ;GENERAL STORAGE (NOT INITIALISED)

4045                 NAME:   DEFS    6               ;TARGET FILE NAME
404B                 PATL:   DEFS    1               ;LENGTH OF PATTERN
404C                 PATN:   DEFS    1               ;START OF PATTERN

        4100         STACK   EQU     RAM+100H

                     ;PORT ADDRESSES

        0FFC         KBD     EQU     0FFCH
        0FFC         PORT0   EQU     0FFCH
        0FFD         CNTR    EQU     0FFDH
        0FFE         PORT1   EQU     0FFEH

                     ;PORT 0 BIT ALLOCATION

        0092         MKINIT  EQU     10010010B       ;INITIAL STATE OF PORT 0 MASK
        0007         PAGE    EQU     7               ;MEMORY PAGE SWITCH (=1 FOR OFF)
        0006         CLRCNT  EQU     6               ;CLEAR 8 USEC COUNTER
        0005         RLY2    EQU     5               ;RELAY 2 (=0 FOR CLOSED)
        0004         FREQ    EQU     4               ;SET 2400 HZ
        0003         RLY1    EQU     3               ;RELAY 1 (=0 FOR CLOSED)
        0002         VTOPN   EQU     2               ;OPEN VT MEMORY
        0001         SINGLE  EQU     1               ;ENABLE SINGLE STEP (=0 FOR SET)
        0000         CLRKBD  EQU     0               ;CLEAR KEYBOARD LATCH

                     ;PORT 1 BIT ALLOCATION

        0007         LINE    EQU     7               ;VT LINE WAVE FORM
        0006         FRAME   EQU     6               ;VT FRAME WAVE FORM
        0005         TAPE    EQU     5               ;CASSETTE SIGNAL SENSE
        0003         VOL     EQU     3               ;CASSETTE VOLUME SENSE
        0002         RESET   EQU     2               ;RESET BUTTON
        0001         CLK     EQU     1               ;1200 HZ CLOCK

        0047         VALUE   EQU     47H             ;TAPE CYCLE DISCRIMINATOR
        0080         RECLEN  EQU     128             ;RECORD LENGTH IN BYTES
        0002         GAP     EQU     2               ;INTER REC GAP (100'S OF MSEC)
        00F0         GAPLEN  EQU     240             ;GAP TO FIND WHEN READING ('1' CYCLES)
        004D         SOH     EQU     01001101B       ;START OF RECORD (4DH)

        000A         LF      EQU     0AH
        000C         FORM    EQU     0CH
        000D         CR      EQU     0DH

        0003         CTRLC   EQU     3
        0015         CTRLU   EQU     15H
```

```
001A        CTRLZ   EQU     1AH

0080        BLANK   EQU     80H             ;GRAPHICS BLANK
005B        LARR    EQU     5BH
005D        RARR    EQU     5DH
005E        UPARR   EQU     5EH
007F        DELETE  EQU     7FH
007F        CURSOR  EQU     7FH

```

```
                    1730  *H SYMBOL TABLE
0000                1731         END
```

SYMBOL TABLE:

...OUTC	0547	.BYTEO	0694	.CHAN	006E	.CHN1	0075
.CHN2	0076	.CLEAR	0513	.CLOSE	0518	.DEOUT	068F
.EDGE	07A1	.GRAFX	0326	.GTBYT	06A8	.GTGAP	0724
.GTHEX	0603	.GTSYN	0715	.KBDIN	04E0	.MSG	05CF
.MSG1	05D9	.OPNWT	04FC	.OUTC	0547	.PTBYT	072E
.SCROL	031A	.UPDAT	0522	ADDR	400B	AOUT	053A
BEGIN	0337	BG1	0347	BG2	0358	BLANK	0080
BREAK	0038	BYTEO	0015	CCLEN	0004	CGTHEX	05FD
CHAN	0018	CHO	06A5	CLAST	05B3	CLEAR	000F
CLK	0001	CLOSE	000C	CLR	052B	CLR1	052E
CLR2	0530	CLRCNT	0006	CLRKBD	0000	CLSOPN	04FA
CMD	01CD	CMD1	01E5	CMD2	01FE	CNTR	0FFD
COMP	03F2	CONTC	0371	CONTIN	0289	CR	000D
CROUT	0545	CTRLC	0003	CTRLU	0015	CTRLZ	001A
CURON	05BD	CURSOR	007F	DELETE	007F	DEOUT	0014
DEOUTC	068A	DEOUTW	0688	ED1	07A8	ED2	07AC
EDGE	0010	ERROR	05F3	ERROUT	05C9	FFLEN	0021
FIND	03CF	FORM	000C	FPJMP	0313	FPM	00D6
FPM1	00DF	FPM2	00E3	FRAME	0006	FREQ	0004
FSTLST	05E2	GAP	0002	GAPLEN	00F0	GB1	06B2
GB2	06B7	GB3	06BD	GB4	06D4	GB5	06D7
GB6	06DD	GB7	06E6	GC1	06F9	GC2	0704
GET	043B	GETBYT	0004	GETC	00CD	GETCYC	06F6
GETGAP	400D	GETHDR	0429	GETHEX	0013	GETNAM	0654
GETPAT	02D7	GETREC	03FA	GETSYN	0011	GG1	0726
GP1	02DB	GP2	02EE	GP3	0307	GPNXT	02EE
GR1	0418	GR2	0426	GRAFBK	07FD	GRAFIX	000B
GRAFNC	032D	GT1	0442	GTSN1	071A	HEXCAL	02BB
HEXO	069D	HIMEM	4043	IN1	4025	IN2	4028
IN3	402B	IND	0255	INIT	031D	INITBK	07DE
INITLN	001F	IT1	0320	JUMP	0316	KBD	0FFC
KBD1	04EF	KBDIN	0002	KBDV	4013	LARR	005B
LDER1	040C	LDERR	0409	LDRET	0092	LF	000A
LINE	0007	LOAD	03C9	LOOKUP	0227	LPV	401C
MASK	4003	MD1	01A8	MD2	01B1	MDIS	0198
MGTHEX	05F8	MKINIT	0092	MONEND	07FF	MONLEN	0800
MONST	0000	MONX	4040	MOUT	05CC	MOVE	03A5
MOVER	0335	MPTR	4007	MS100	0798	MSG	0017
MSGB	000B	MSGE	0013	MSGF	001B	MSGL	002B
MSGM	0006	MSGN	07D1	MSGP	0034	MSGS	0623
MSGV	07D6	MV1	03C3	MV2	03C6	NAME	4045
NLINES	4000	NM1	0662	NM2	0673	NM3	0674
NM4	067B	NMI	00A2	NMI1	00AF	NMI1A	00B7
NMI2	00B8	NMIX	403A	NXTC	0607	NXTC1	0623
NXTC2	0631	NXTC3	0637	NXTC4	0640	NXTC5	064E
NXTROM	1000	OPNW1	0501	OPNW2	0505	OPNWT	000B
OUT1	401F	OUT2	4022	OUTC	0016	OUTC1	0560
OUTC1A	0569	OUTC2	056C	OUTC3	0573	OUTC5	057E
PAGE	0007	PATL	404B	PATN	404C	PBT1	075C

3-23.40

PBT2	0764	PBT3	0768	PBT4	076F	PBY1	073A
PBY2	0743	PGTHEX	0600	PMTOUT	0541	PORTO	OFFC
PORT1	OFFE	PR1	04CA	PR2	04D0	PT1	02A5
PT2	02B8	PUT	0297	PUTBIT	0746	PUTBYT	0003
PUTREC	04B2	PUTRNA	04B4	RA1	026D	RAM	4000
RARR	005D	RC1	0783	RCALL	0085	RD1	0118
RD2	0124	RD3	013E	RD4	0153	RD5	015B
RD6	0165	RDCNT	0773	RDIS	00F3	RECLEN	0080
RECNO	044C	REL	025E	RESCNT	0786	RESET	0002
RET1	05C4	RET2	05C5	RLY1	0003	RLY2	0005
RN1	017C	RNAM	0174	RNSTR	0180	RNTR	039E
RPTR	4009	RSET	400A	RST10	4031	RST18	4034
RST20	4037	RST8	402E	RTOA	0265	SAVE	0462
SCRL	058A	SCRL1	059B	SCROLL	000E	SEC5	0796
SETM	0250	SETR	0208	SINGLE	0001	SOH	004D
SR1	0219	SR2	021F	SSFLG	4004	STACK	4100
STEP	0281	SV1	0489	SV2	049A	SV3	04A4
SVM	01FA	SWITCH	0270	TAPE	0005	TBL1	0230
TBL2	0238	TBL2LN	0018	TI	0445	TIV	4019
TO	04DB	TOP	4001	TOV	4016	TR1	0068
TR2	007A	TR3	0082	TRAP	0049	TRAPX	403D
TRTBL	07B5	UPARR	005E	UPD1	0523	UPDATE	0012
VALUE	0047	VOL	0003	VTOPN	0002	VTPTR	4005
VTV	4010	ZMON	0379				

NO ERRORS